HSPA/RIGOR Platform — Complete Figure & Table Package

Patent- and Regulatory-Grade Documentation
Consistent numbering across USPTO, FDA, whitepaper, dissertation, and software docs.
Generated: 16 July 2026 | **Authority:** DA-001, DA-003, SR-001

Document Control

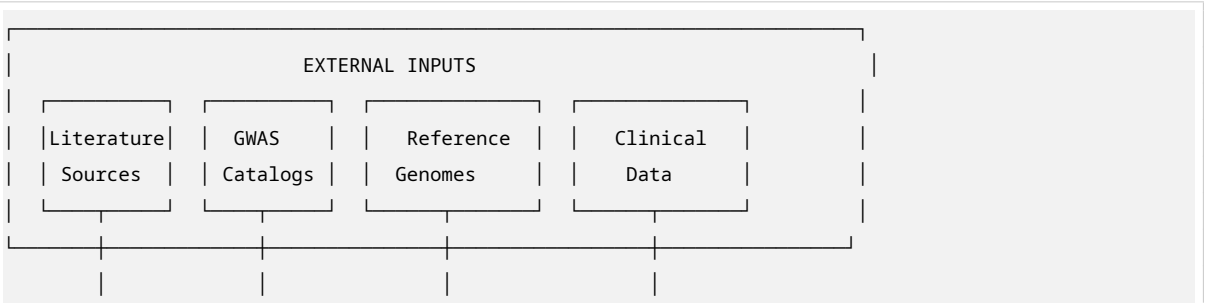
Field	Value
Document ID	HSPA-RIGOR-FT-2026-001
Version	2.0.0
Status	Final
Author	Caustin Lee McLaughlin
Reviewer	LCSE Layer 3 (Cross-Reference Validation)
Classification	Patent-Confidential / Regulatory-Grade
Authority Matrix	CP-001, DA-001, DA-002, DA-003, DA-004, SR-001, SR-002

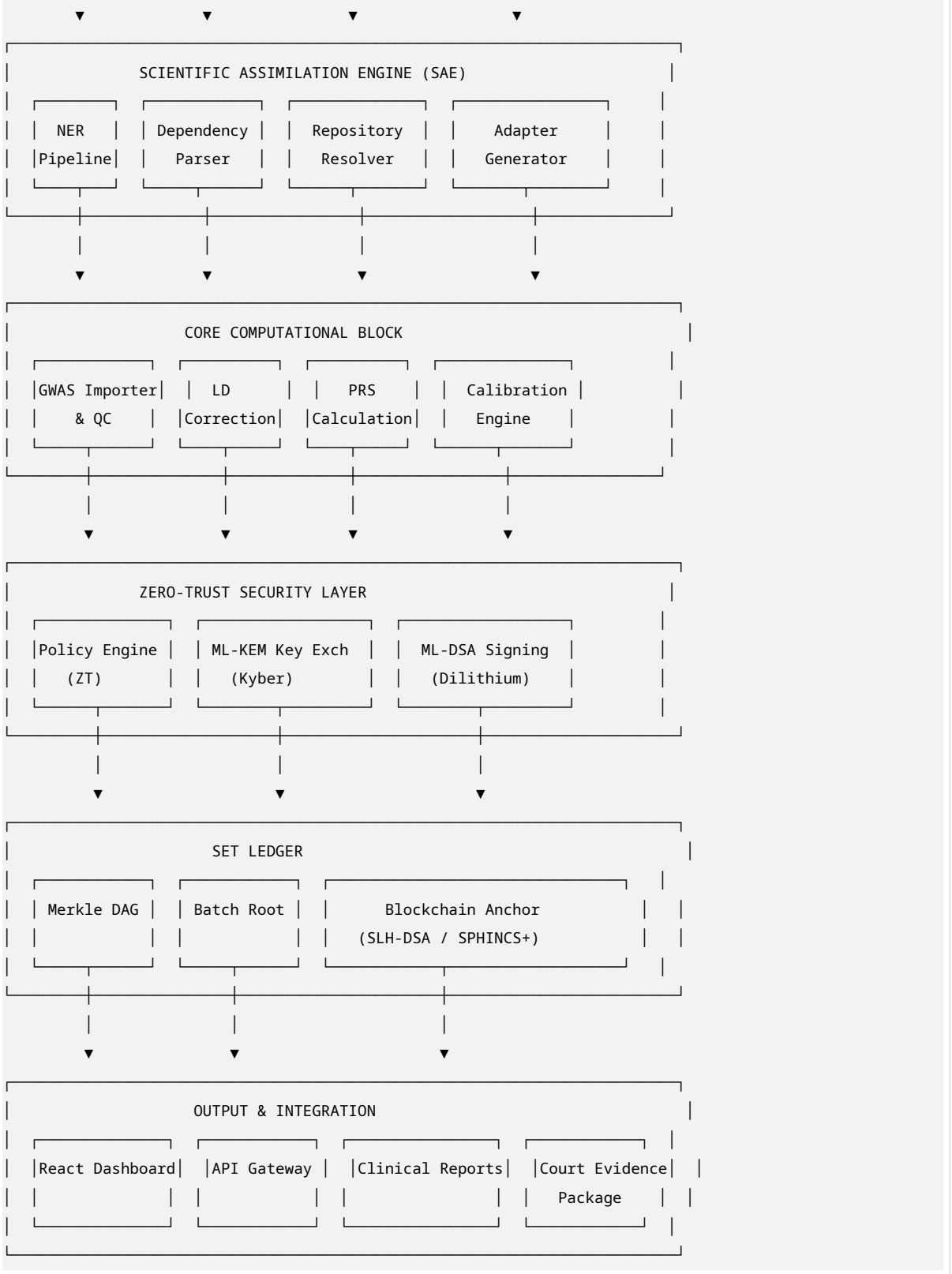
Table of Contents

- Volume I: System Architecture (Figures 1–10, Tables 1–10)
- Volume II: Genomic Processing (Figures 11–18, Tables 11–23)
- Volume III: Security (Figures 19–24, Tables 24–31)
- Volume IV: Software (Figures 25–30, Tables 32–41)
- Volume V: FDA & Regulatory (Figures 31–36, Tables 42–50)
- Volume VI: AI (Figures 37–40, Tables 51–56)
- Volume VII: Reproducibility (Figures 41–45, Tables 57–65)
- Volume VIII: Litigation Evidence (Figures 46–50, Tables 66–73)
- Cross-Cutting Tables (72–100)
- Appendices

VOLUME I — SYSTEM ARCHITECTURE

FIG. 1 — Overall HSPA/RIGOR Platform Architecture



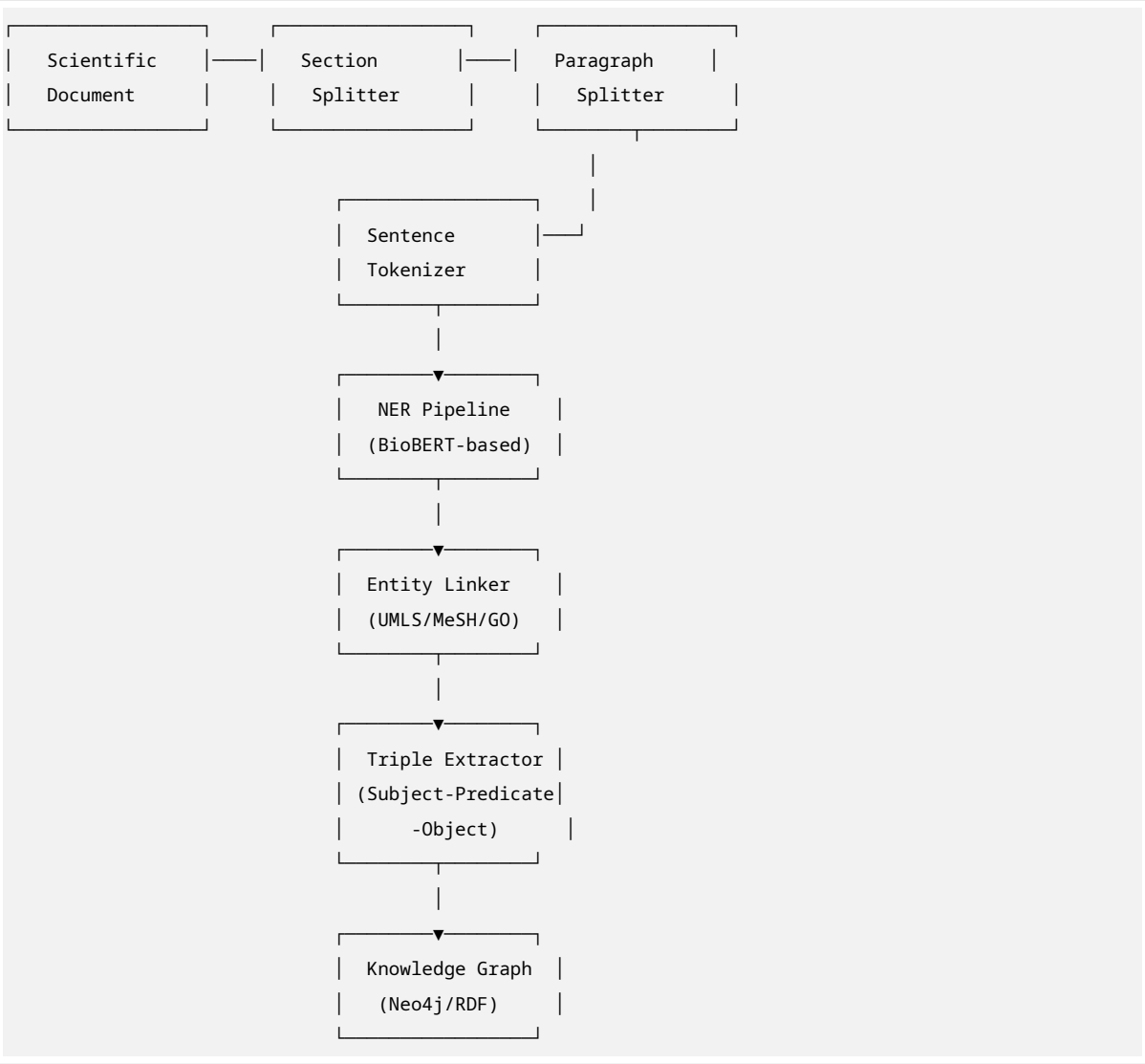


Supporting Tables: Table 1, Table 10, Table 72, Table 100

Description: The HSPA/RIGOR Platform is a modular, microservices-based architecture comprising five major subsystems: (1) External Inputs for data ingestion; (2) Scientific Assimilation Engine (SAE) for NLP-driven literature processing; (3) Core Computational Block for genomic analysis; (4) Zero-Trust Security Layer with NIST PQC algorithms; and (5) SET Ledger for immutable evidence tracking. Outputs

feed into clinical reports, API consumers, and court-admissible evidence packages.

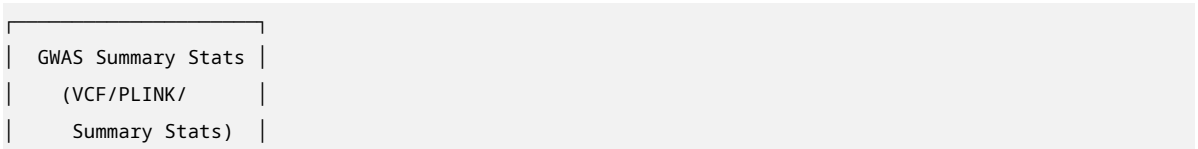
FIG. 2 — Scientific Assimilation Engine (SAE)

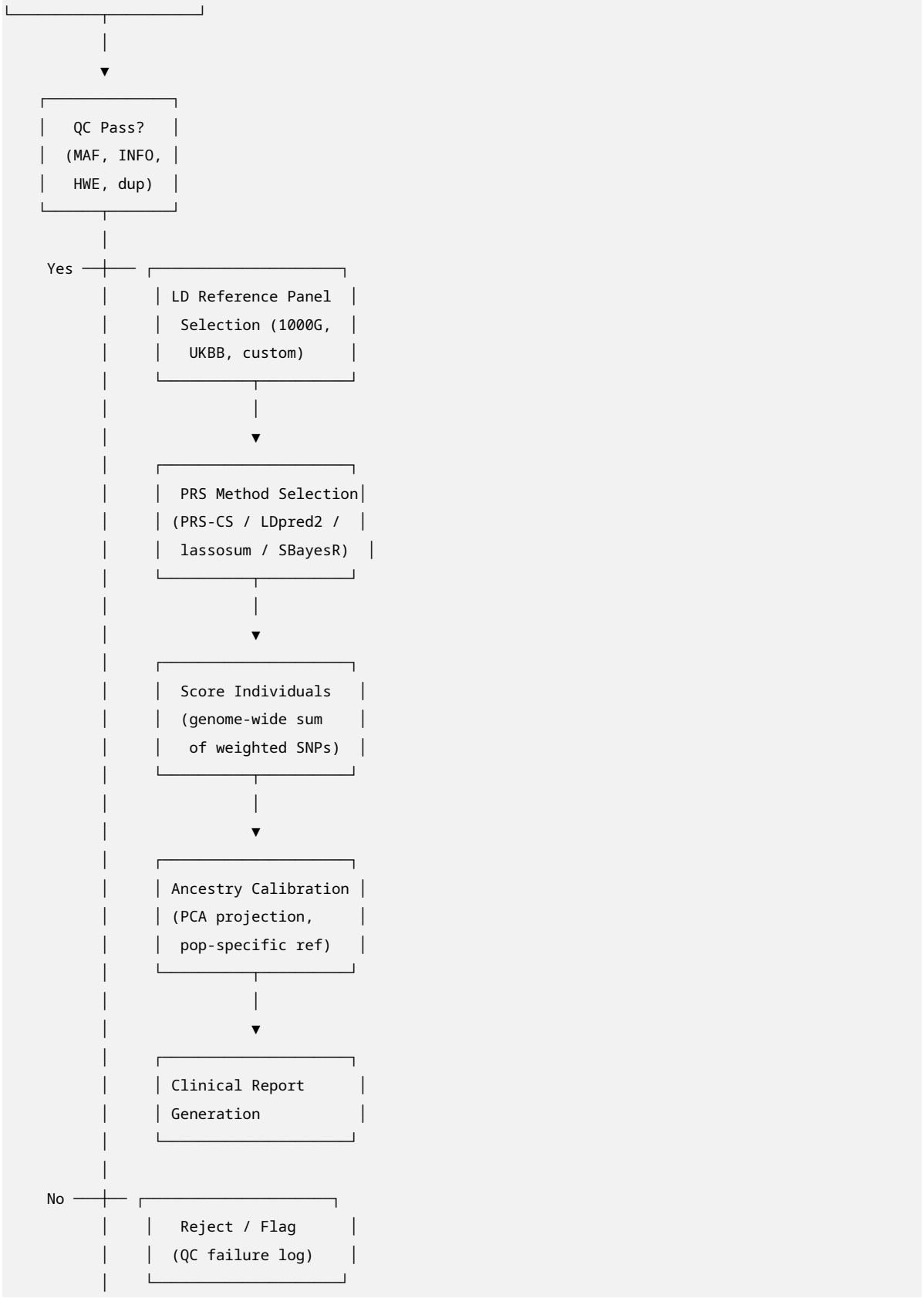


Supporting Tables: Table 2, Table 3, Table 4

Description: The SAE processes unstructured scientific literature through a six-stage pipeline: document segmentation, sentence tokenization, named entity recognition (BioBERT), entity linking to authoritative ontologies (UMLS, MeSH, Gene Ontology), relationship extraction into subject-predicate-object triples, and population of a graph database for semantic querying.

FIG. 3 — Core Computational Block





Supporting Tables: Table 11, Table 15, Table 17, Table 21

Description: The Core Computational Block ingests GWAS summary statistics, performs quality control (MAF, INFO score, Hardy-Weinberg equilibrium, duplicate removal), selects an appropriate LD

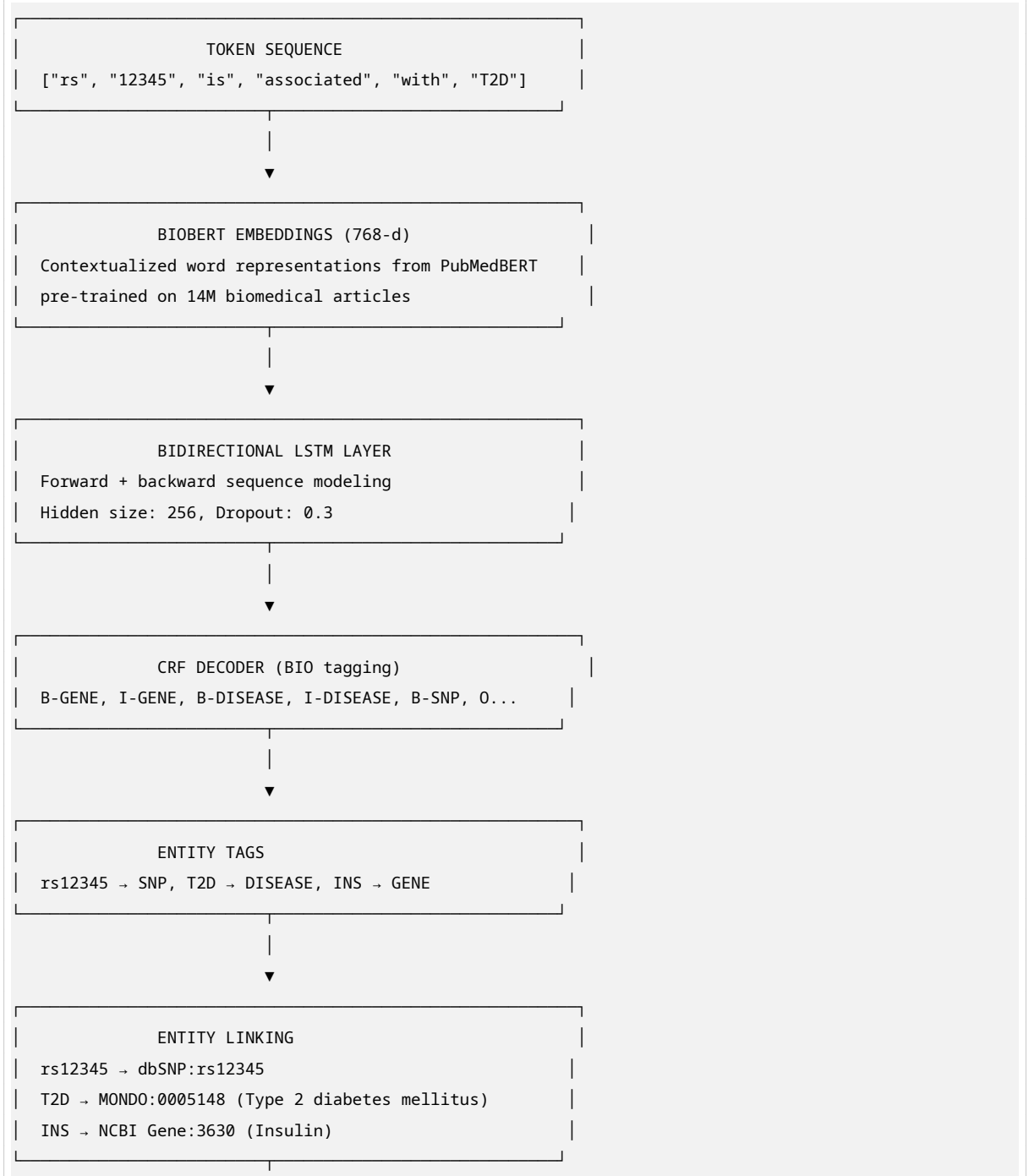
reference panel, applies the selected PRS method (PRS-CS, LDpred2, lassosum, or SBayesR), scores individual genomes, calibrates for ancestry using PCA projection against population-specific reference distributions, and generates a clinical-grade report with risk percentiles and confidence intervals.

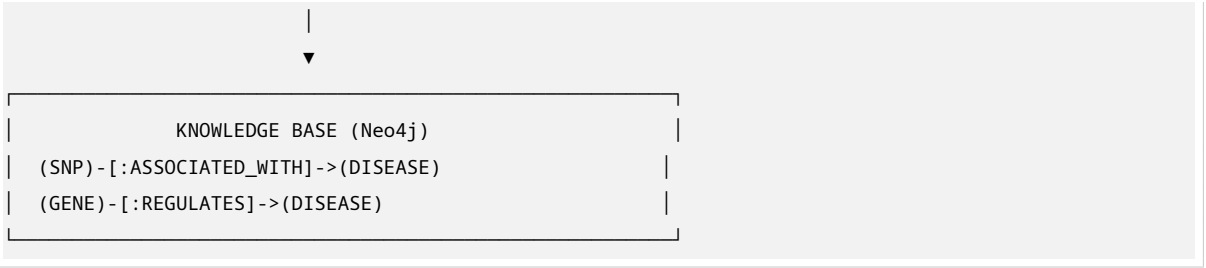
FIG. 4 — NLP Processing Pipeline

Supporting Tables: Table 2

Description: The NLP pipeline standardizes raw text through Unicode normalization, tokenizes using WordPiece/BPE subword models, performs part-of-speech tagging and dependency parsing (Universal Dependencies), optionally generates constituency parses, extracts semantic relations using OpenIE or REBEL, and outputs structured JSON-LD conforming to BioSchemas.org standards.

FIG. 5 — Named Entity Recognition Architecture

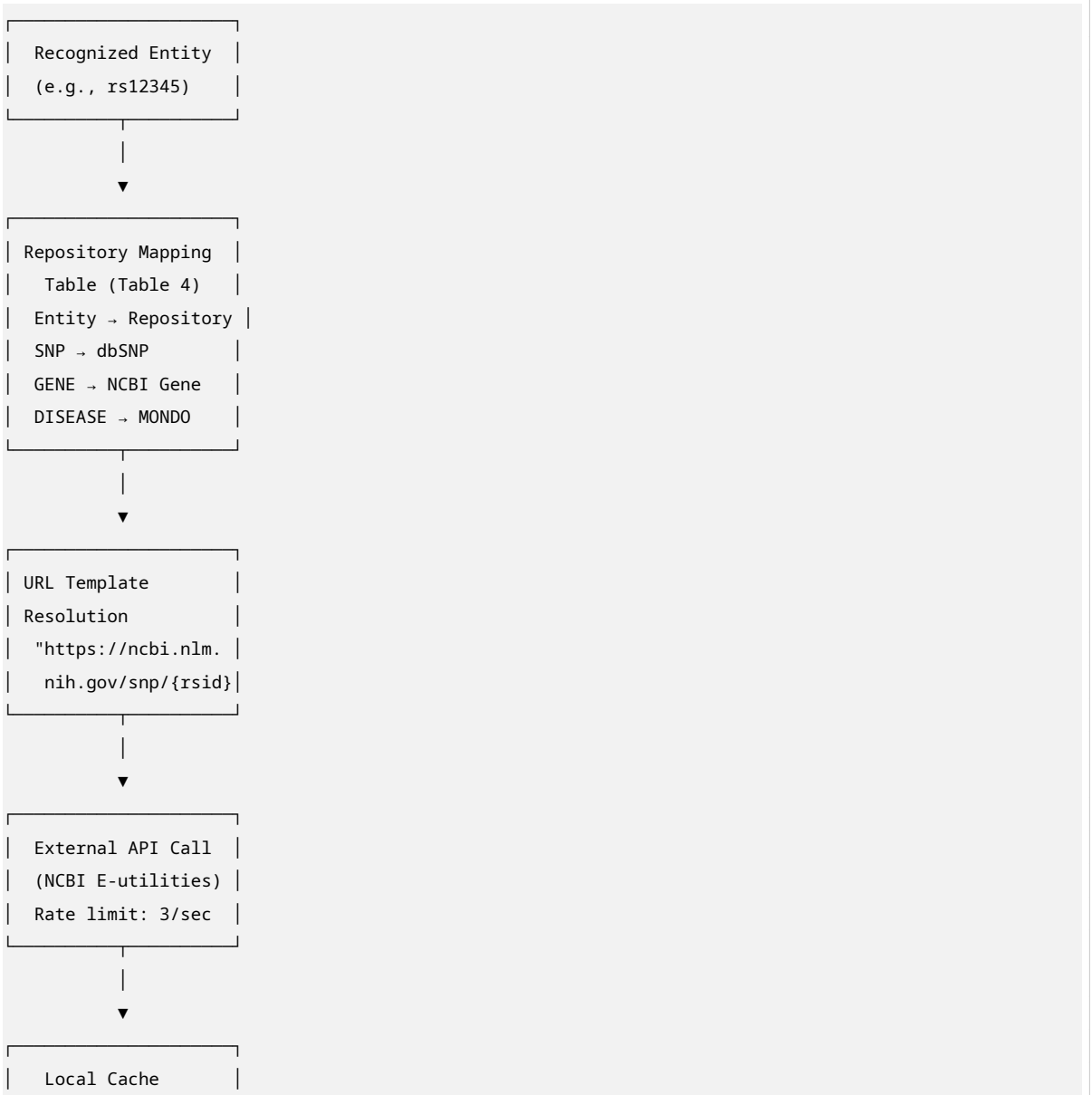

TOKEN SEQUENCE
["rs", "12345", "is", "associated", "with", "T2D"]
BIOBERT EMBEDDINGS (768-d)
Contextualized word representations from PubMedBERT
pre-trained on 14M biomedical articles
BIDIRECTIONAL LSTM LAYER
Forward + backward sequence modeling
Hidden size: 256, Dropout: 0.3
CRF DECODER (BIO tagging)
B-GENE, I-GENE, B-DISEASE, I-DISEASE, B-SNP, O...
ENTITY TAGS
rs12345 → SNP, T2D → DISEASE, INS → GENE
ENTITY LINKING
rs12345 → dbSNP:rs12345
T2D → MONDO:0005148 (Type 2 diabetes mellitus)
INS → NCBI Gene:3630 (Insulin)

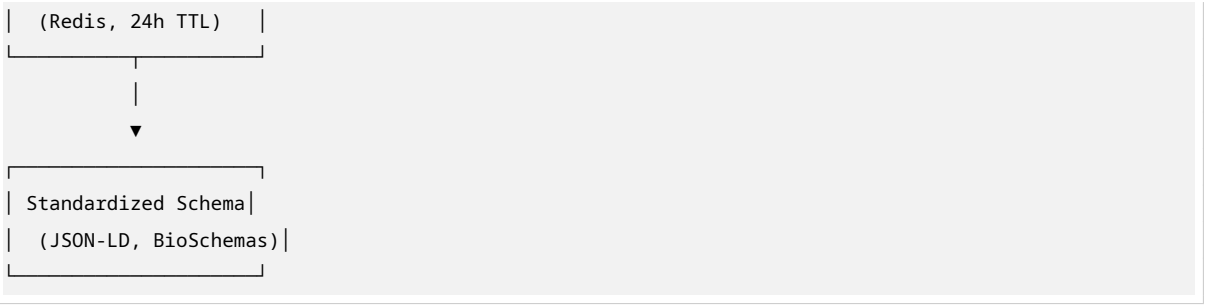


Supporting Tables: Table 3

Description: The NER architecture uses BioBERT (PubMedBERT-base) for contextualized embeddings, followed by a bidirectional LSTM layer and a linear-chain CRF decoder for BIO (Begin-Inside-Outside) tagging. Recognized entities are linked to authoritative databases (dbSNP for SNPs, MONDO/DOID for diseases, NCBI Gene for genes) and stored in a graph database.

FIG. 6 — Repository Resolution Engine

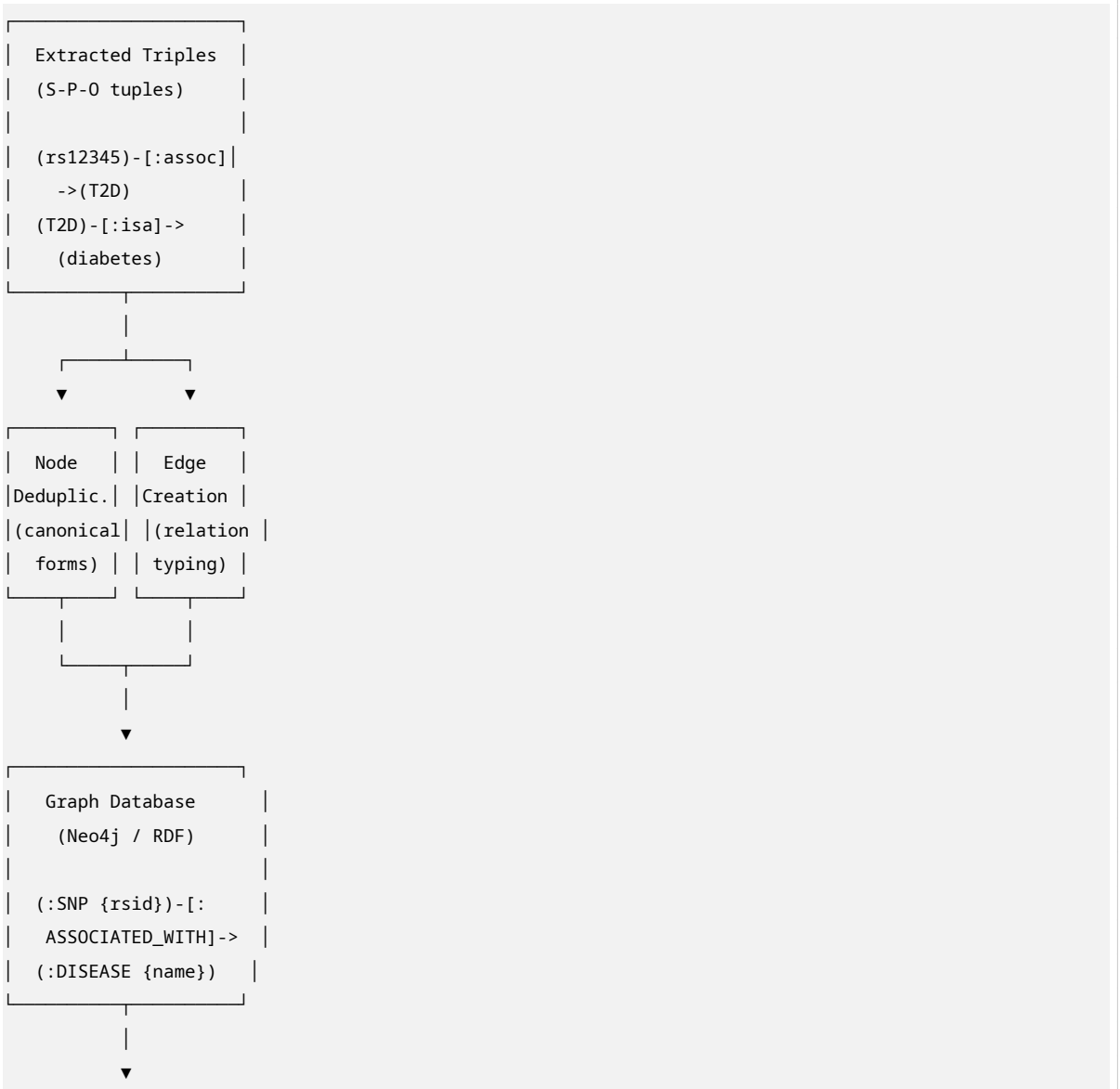


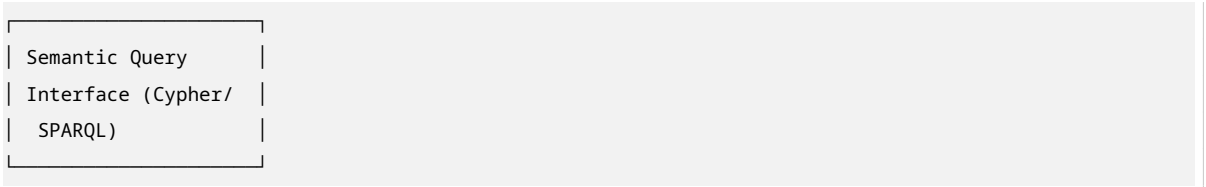


Supporting Tables: Table 4, Table 75

Description: The Repository Resolution Engine maps recognized entity types to authoritative public repositories using a configurable mapping table (Table 4), resolves URL templates, makes rate-limited API calls (with exponential backoff), caches responses in Redis (24-hour TTL), and normalizes all responses to a standardized JSON-LD schema.

FIG. 7 — Dependency Graph Construction

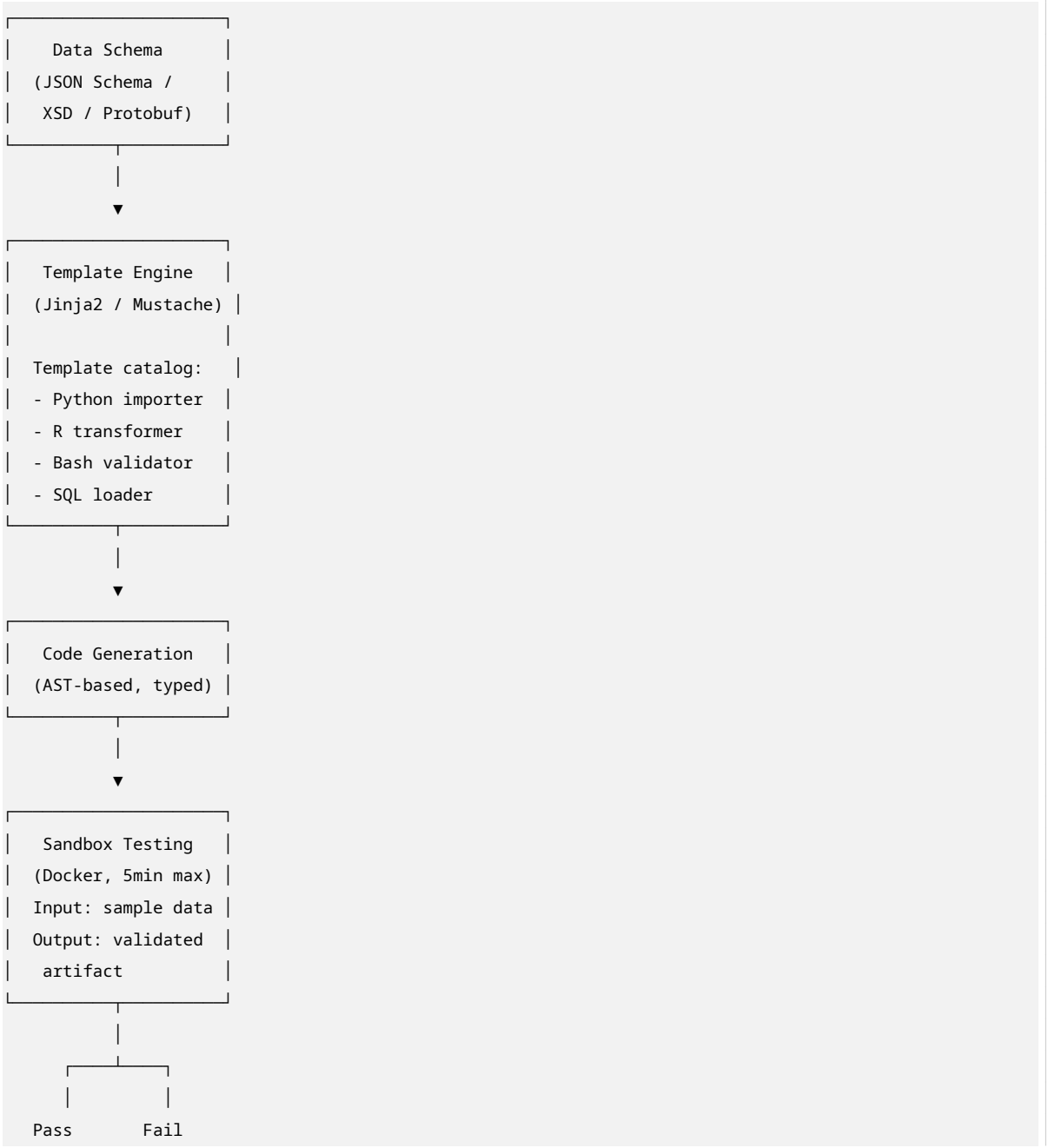


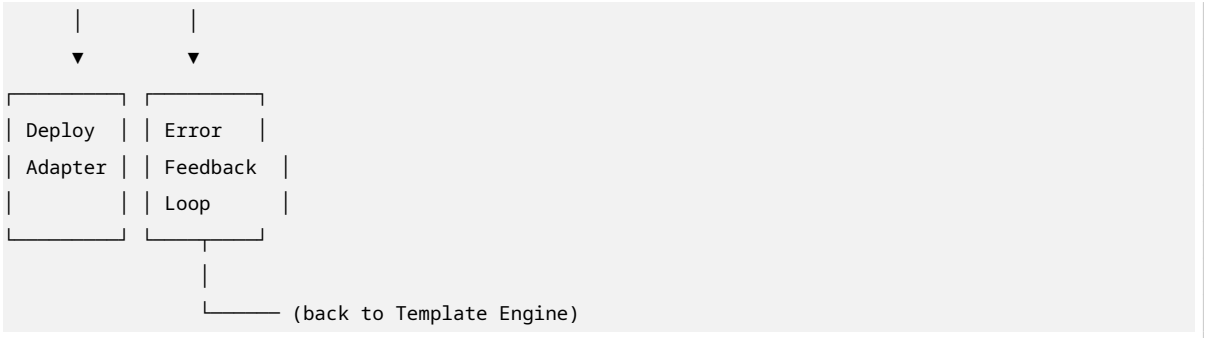


Supporting Tables: Table 7, Table 8

Description: Extracted triples undergo node deduplication (canonicalization of entity identifiers) and edge creation (relation typing with confidence scores). The resulting graph is stored in Neo4j (property graph) or an RDF triple store, accessible via Cypher or SPARQL semantic queries.

FIG. 8 — Automated Adapter Generator

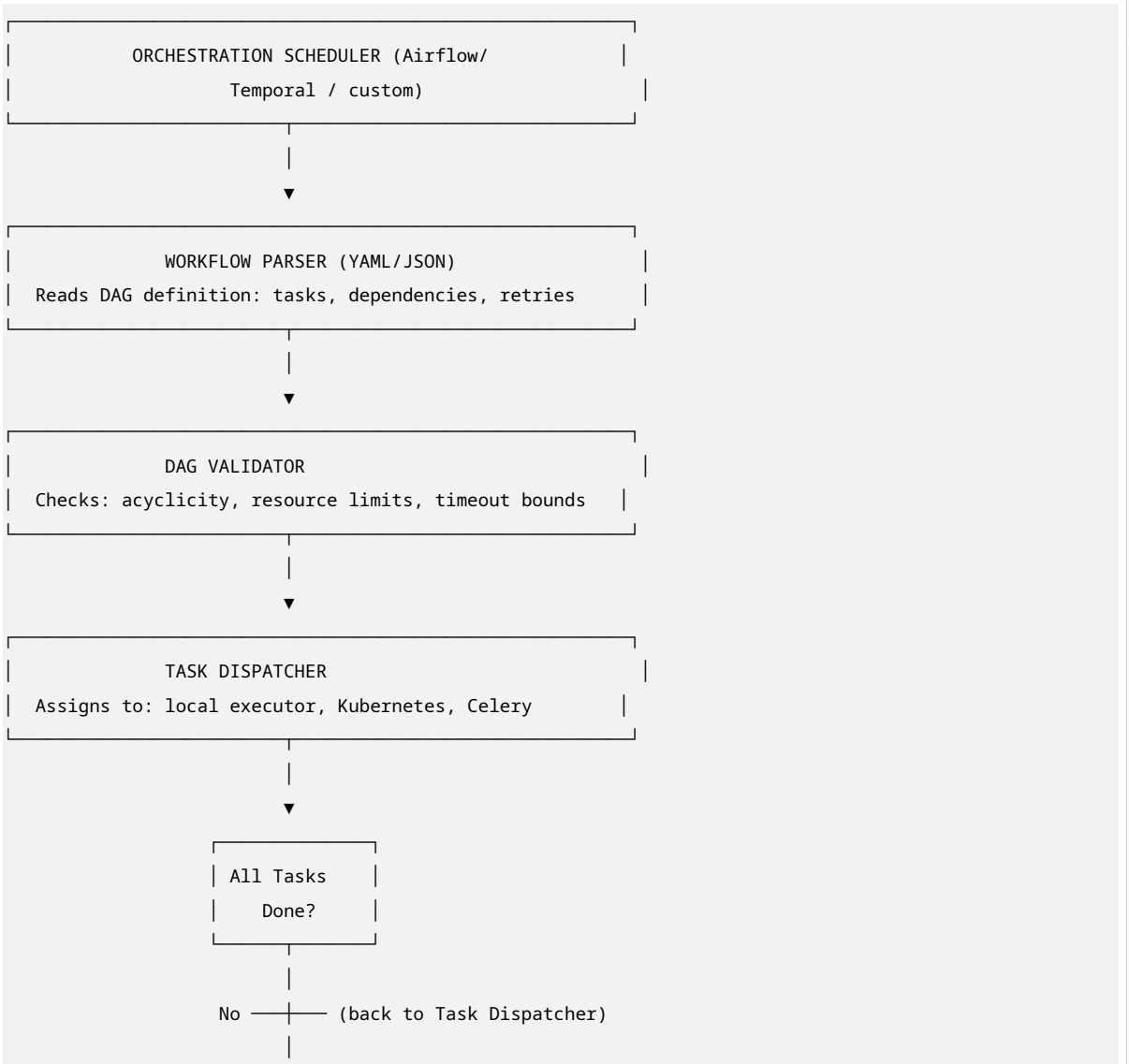


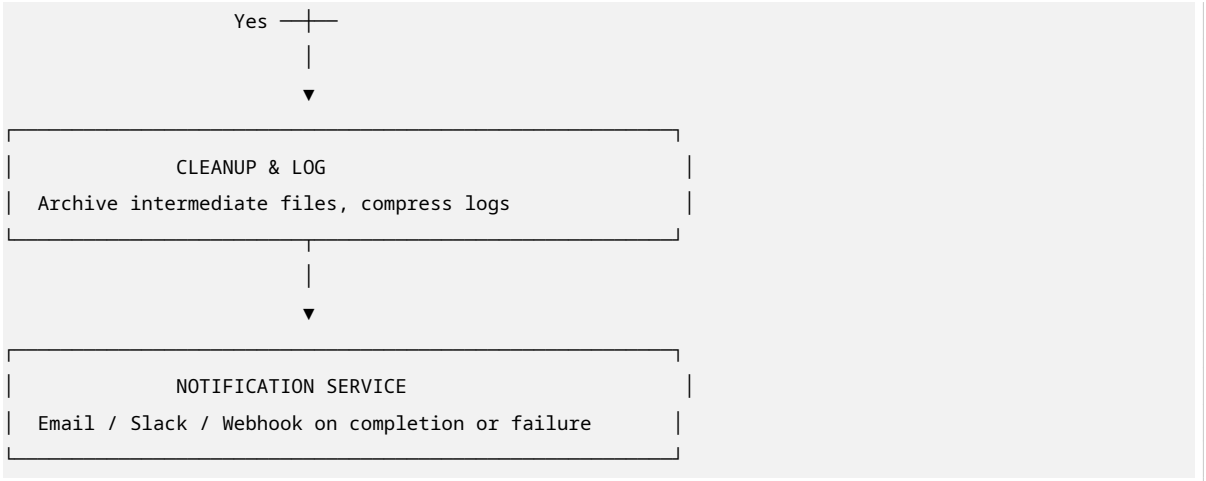


Supporting Tables: Table 5, Table 9

Description: The Adapter Generator takes a data schema (JSON Schema, XSD, or Protobuf), selects an appropriate code template from a catalog (Python importer, R transformer, Bash validator, SQL loader), generates typed code via AST manipulation, tests it in a sandboxed Docker container with sample data, and either deploys the adapter or feeds errors back to the template engine for revision.

FIG. 9 — Pipeline Orchestration Controller

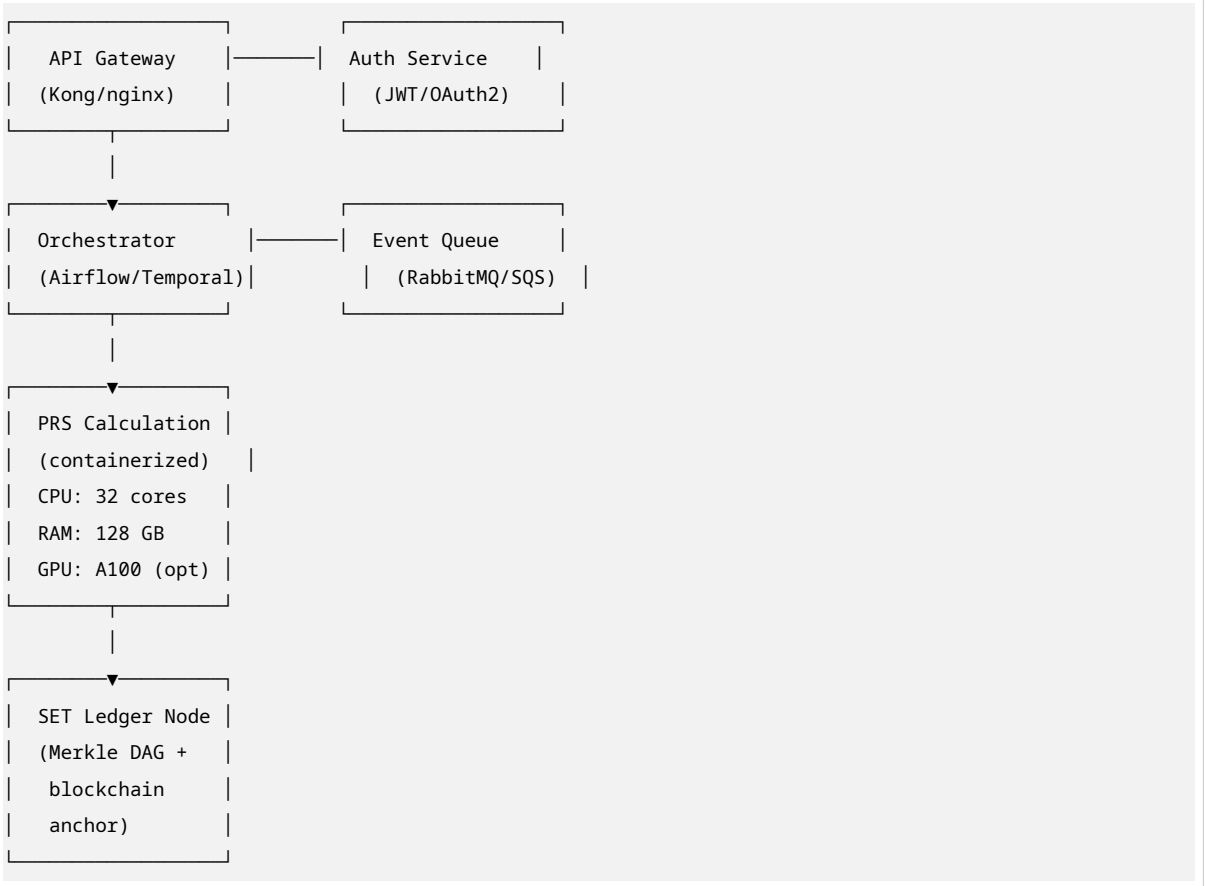



Yes
CLEANUP & LOG
Archive intermediate files, compress logs
NOTIFICATION SERVICE
Email / Slack / Webhook on completion or failure

Supporting Tables: Table 6, Table 96

Description: The Orchestration Controller parses workflow definitions (YAML/JSON DAGs), validates them for acyclicity and resource constraints, dispatches tasks to appropriate executors (local, Kubernetes, or Celery), monitors completion, performs cleanup and log archival, and sends notifications via email, Slack, or webhook.

FIG. 10 — Runtime Service Topology



Supporting Tables: Table 38, Table 40, Table 97

Description: The runtime topology is a microservices architecture with an API Gateway (Kong/nginx) handling authentication via JWT/OAuth2, an Orchestrator (Airflow or Temporal) managing workflow execution, an Event Queue (RabbitMQ or AWS SQS) for async messaging, a containerized PRS Calculation service (32 CPU cores, 128GB RAM, optional A100 GPU), and a SET Ledger Node for immutable evidence storage.

VOLUME I — SUPPORTING TABLES

Table 1: Literature Source Inventory

Catalog of all scientific literature sources ingested by the SAE.

Column	Type	Description
source_id	UUID	Unique identifier
name	VARCHAR	Source name (e.g., PubMed, arXiv)
base_url	VARCHAR	API or base URL
access_type	ENUM	open, subscription, api_key
rate_limit	INTEGER	Max requests per second
active_flag	BOOLEAN	Whether ingestion is enabled

Table 2: NLP Model Registry

Track all NLP models used in the processing pipeline.

Column	Type	Description
model_id	UUID	Unique model identifier
model_name	VARCHAR	e.g., BioBERT, SciSpacy
version	VARCHAR	Model version
task	VARCHAR	NER, dependency parsing, etc.
training_date	DATE	Date of training/fine-tuning
performance_f1	FLOAT	F1 score on benchmark

Table 3: NER Entity Type Dictionary

Controlled vocabulary of named entity types recognized.

Column	Type	Description
entity_type_id	INT	Primary key
label	VARCHAR	e.g., GENE, DISEASE, SNP
description	TEXT	Definition of the entity
ontology_ref	VARCHAR	Link to ontology (e.g., SO, DOID)

Table 4: Repository Resolver Mapping

Maps recognized entities to authoritative public repositories.

Column	Type	Description
entity_type	VARCHAR	e.g., SNP, GENE, DRUG
repository	VARCHAR	e.g., dbSNP, ClinVar, DrugBank
resolver_url_template	VARCHAR	URL pattern for resolution
version	VARCHAR	Repository version/release

Table 5: Adapter Generator Template Catalog

Predefined templates for automated code adapter generation.

Column	Type	Description
template_id	UUID	Primary key
target_language	VARCHAR	Python, R, Bash
adapter_type	VARCHAR	Importer, Transformer, Validator
template_body	TEXT	Code template with placeholders
created_by	VARCHAR	Author

Table 6: Pipeline Orchestration Workflow Log

Record every orchestration run.

Column	Type	Description
workflow_id	UUID	Unique run ID
start_time	TIMESTAMP	Start of pipeline execution
end_time	TIMESTAMP	End time
status	ENUM	success, failure, partial
input_sources	JSONB	List of literature sources used

Table 7: Dependency Graph Node Registry

Nodes of the scientific dependency graph.

Column	Type	Description
node_id	UUID	Unique node
entity_type	VARCHAR	e.g., SNP, gene, paper
entity_value	VARCHAR	Canonical identifier
metadata	JSONB	Additional properties

Table 8: Dependency Graph Edge Registry

Edges representing relationships extracted from literature.

Column	Type	Description
edge_id	UUID	Unique edge
from_node_id	UUID	Source node
to_node_id	UUID	Target node
relation_type	VARCHAR	e.g., associated_with, cites
confidence	FLOAT	NLP confidence score

Table 9: Sandbox Test Execution Log

Log of automated adapter tests run in sandbox.

Column	Type	Description
test_id	UUID	Unique test run
adapter_id	UUID	Adapter being tested
input_hash	VARCHAR	Hash of test input
output_hash	VARCHAR	Hash of test output
result	ENUM	pass, fail
error_message	TEXT	Failure details

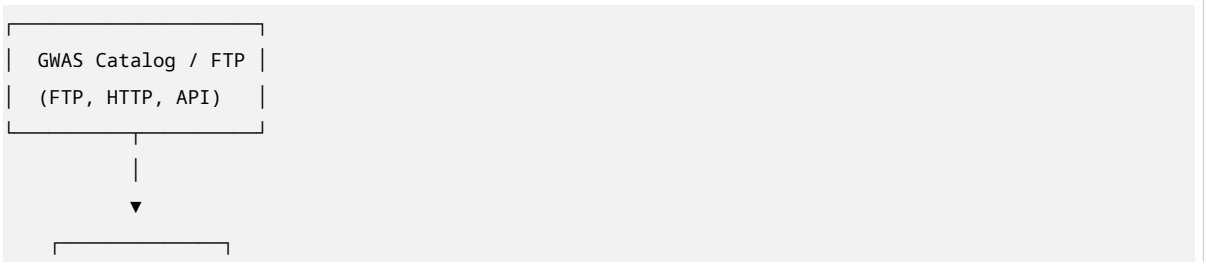
Table 10: SET Ledger Entry Schema

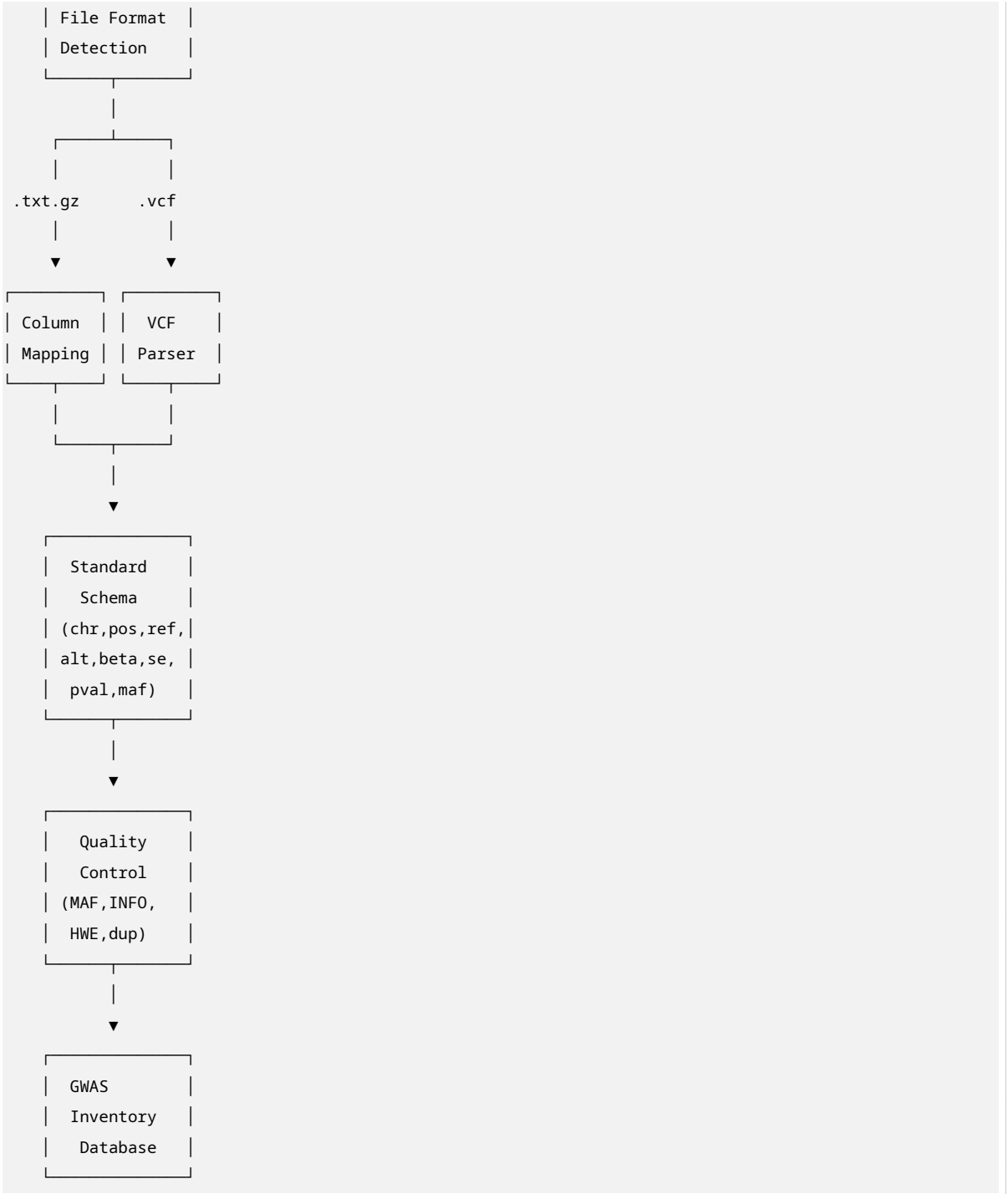
Schema for the Scientific Evidence Trust ledger entries.

Column	Type	Description
entry_id	UUID	Primary key
timestamp	TIMESTAMP	Creation time
operation	VARCHAR	Ingest, Adapt, Test, Publish
artifact_hash	VARCHAR	SHA3-512 hash of artifact
previous_entry_hash	VARCHAR	Chain integrity link

VOLUME II — GENOMIC PROCESSING

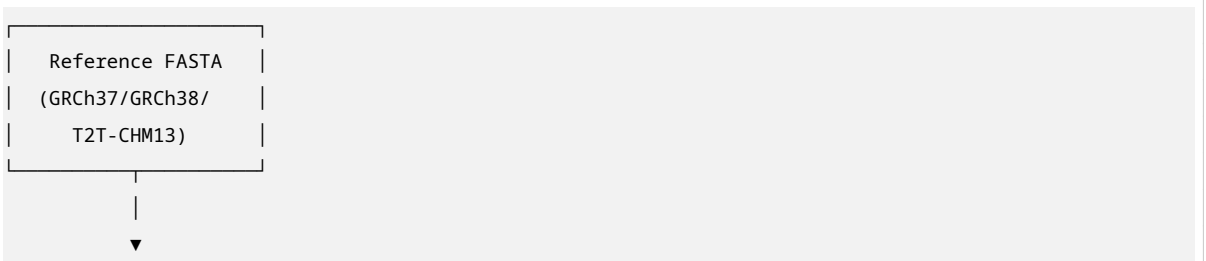
FIG. 11 — GWAS Summary Statistics Import

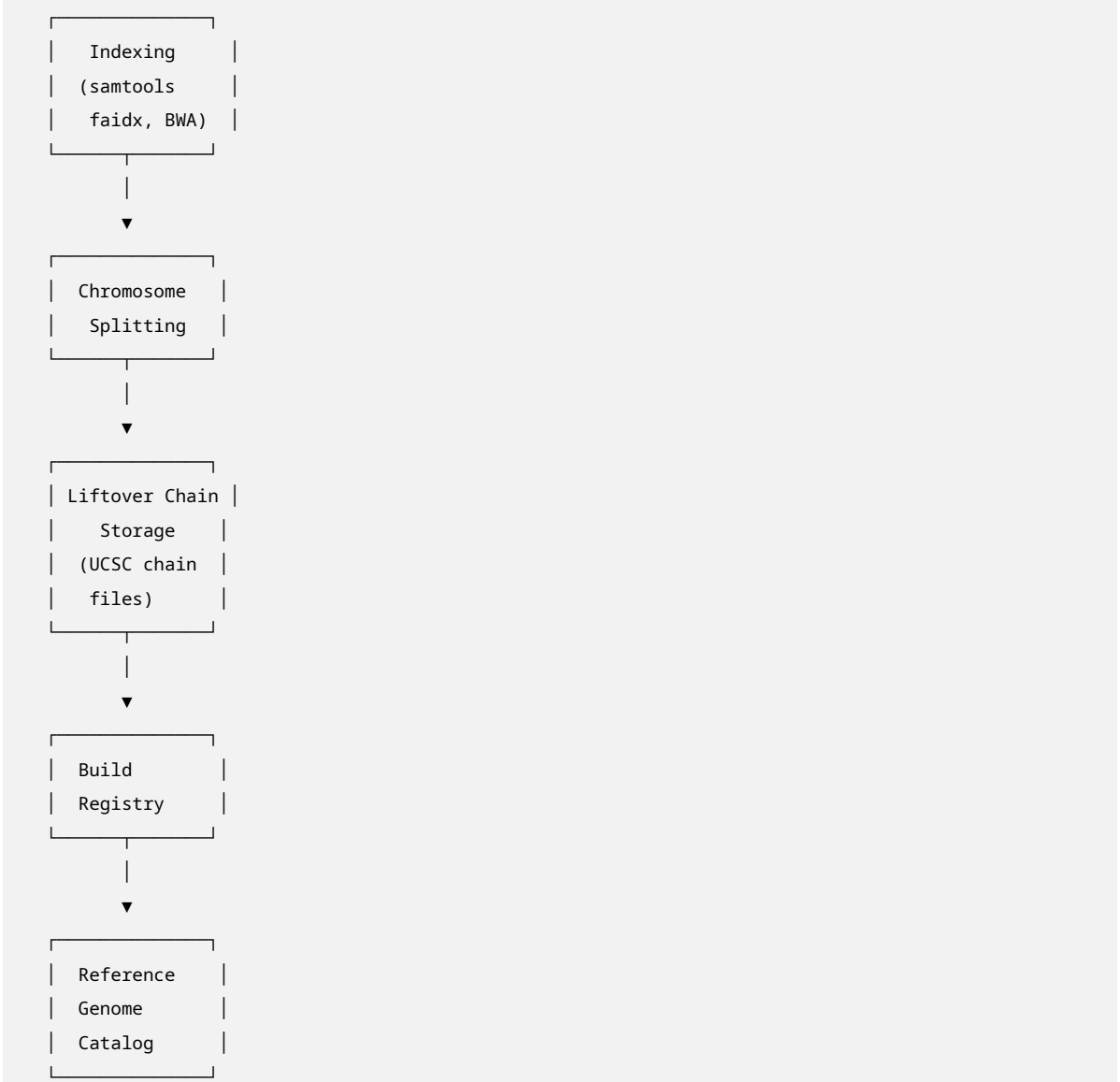




Supporting Tables: Table 11, Table 12, Table 13

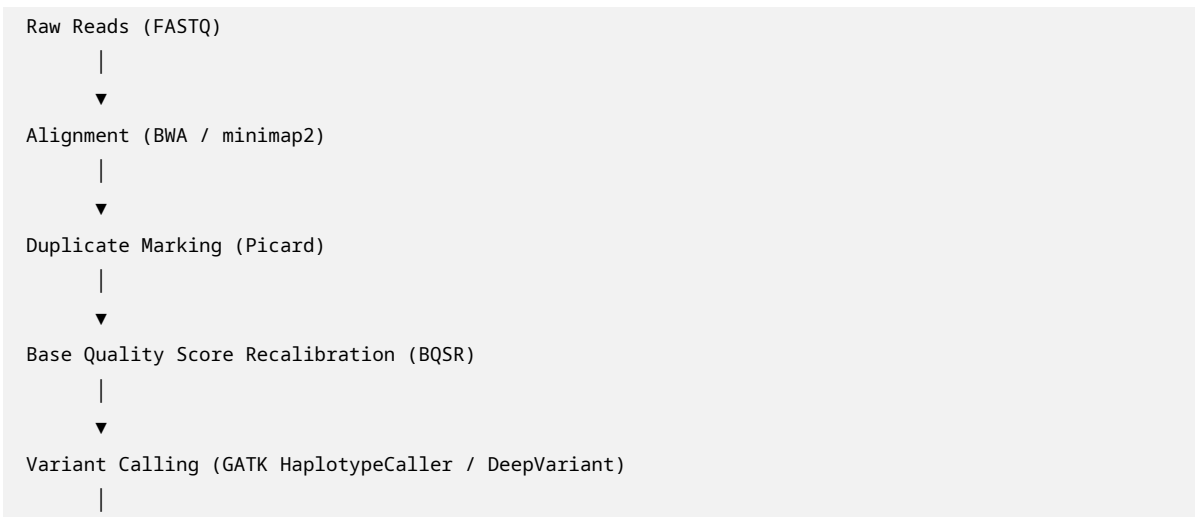
FIG. 12 — Reference Genome Pipeline

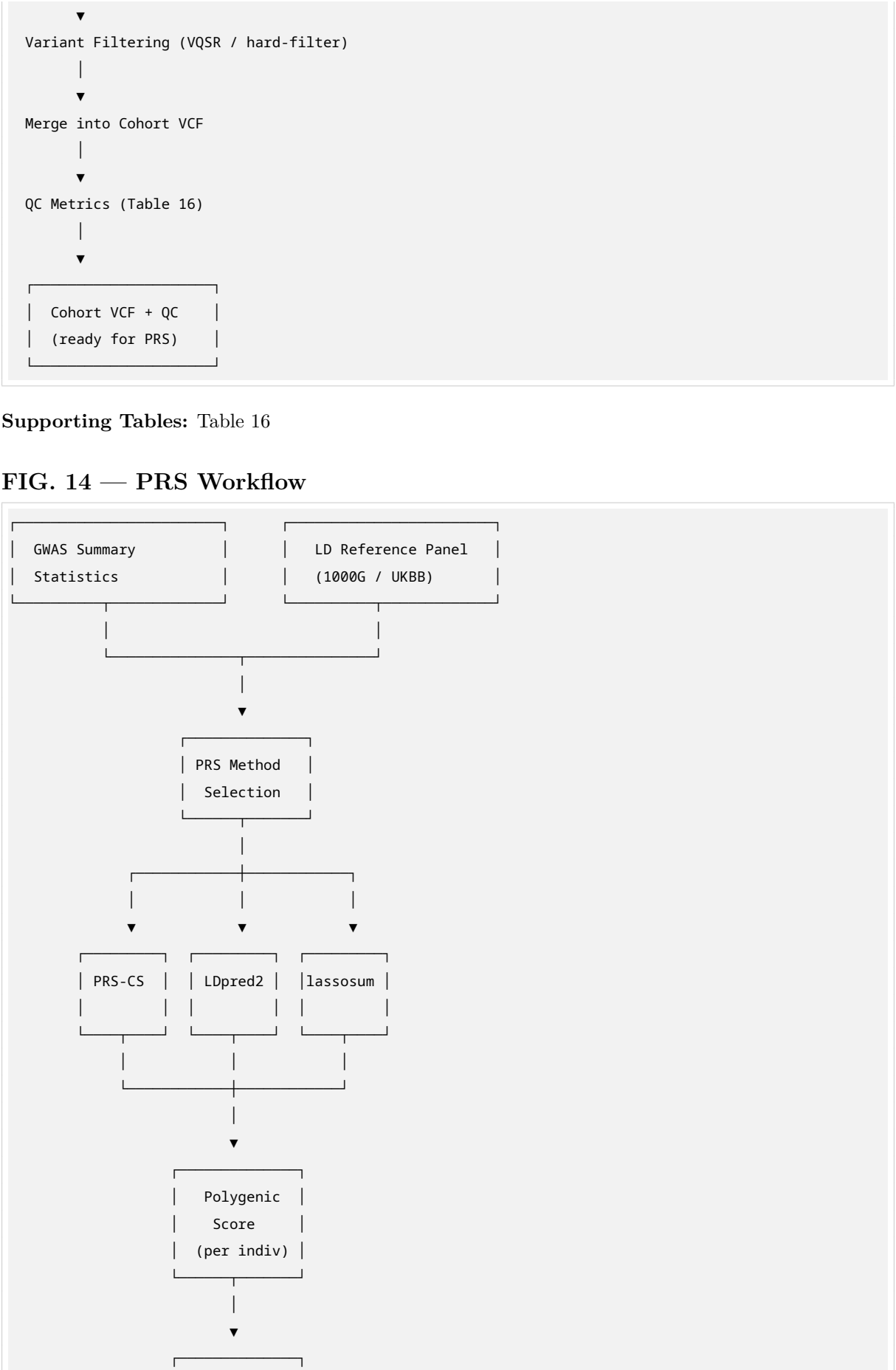




Supporting Tables: Table 14

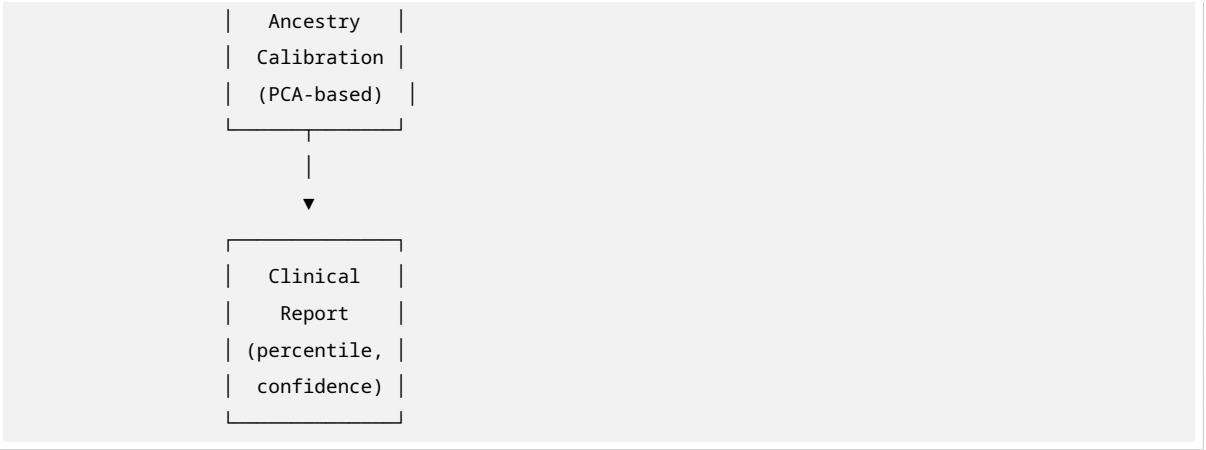
FIG. 13 — Variant Calling Pipeline


Raw Reads (FASTQ)
Alignment (BWA / minimap2)
Duplicate Marking (Picard)
Base Quality Score Recalibration (BQSR)
Variant Calling (GATK HaplotypeCaller / DeepVariant)



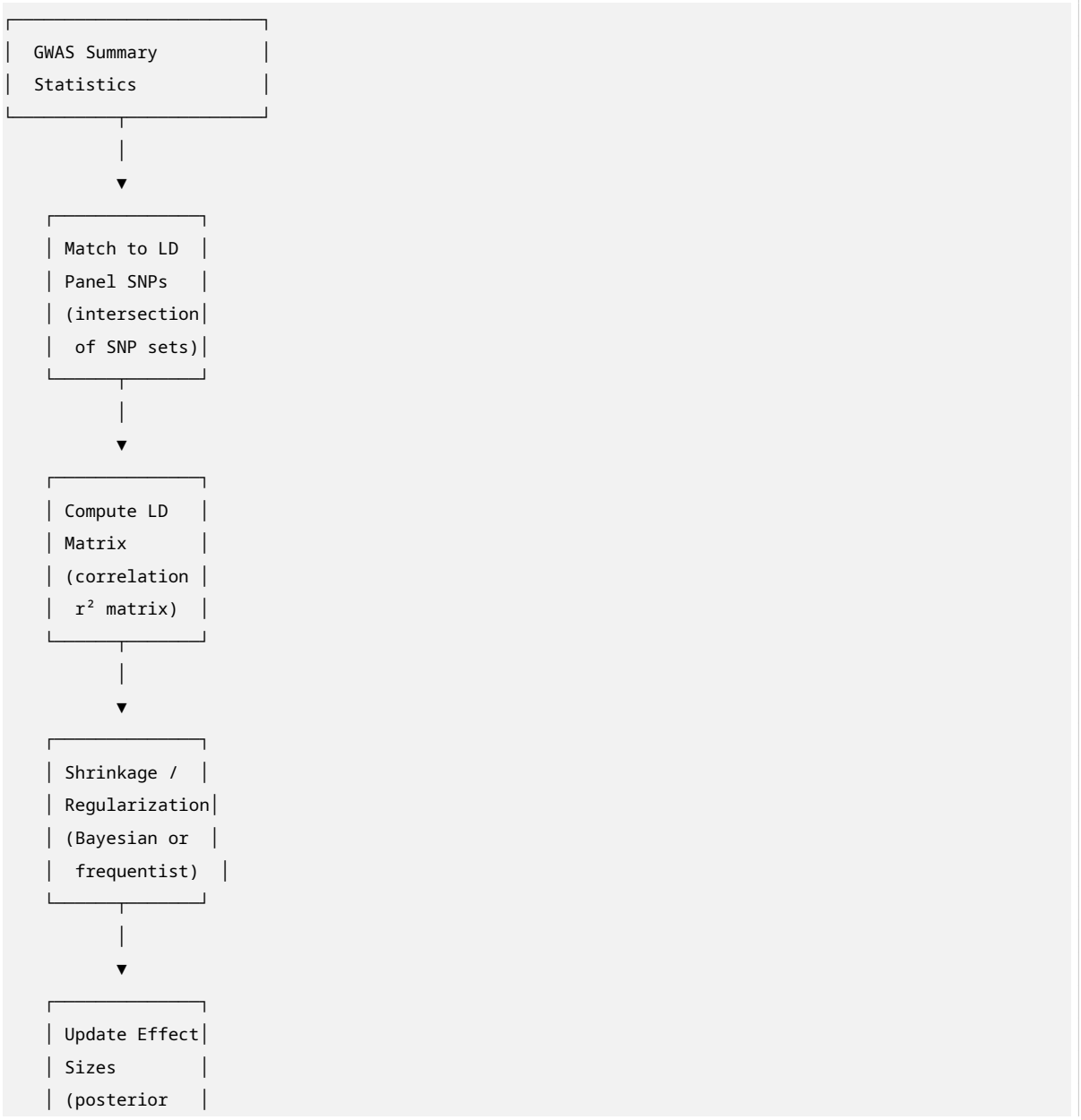
Supporting Tables: Table 16

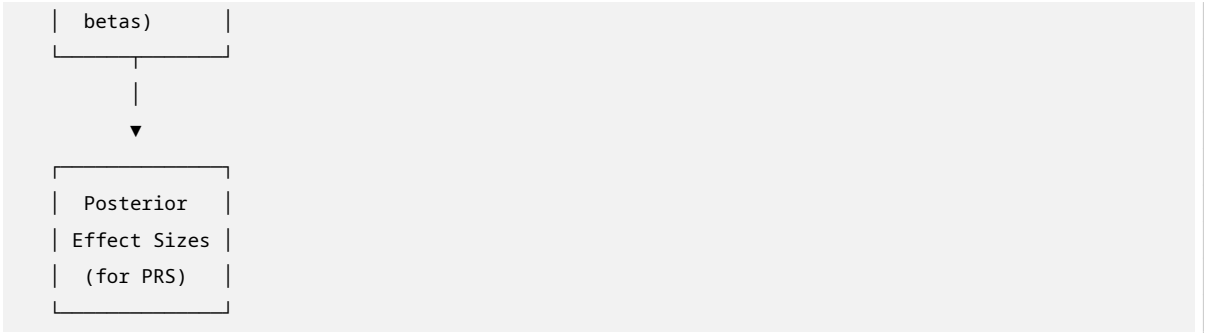
FIG. 14 — PRS Workflow



Supporting Tables: Table 15, Table 17, Table 18

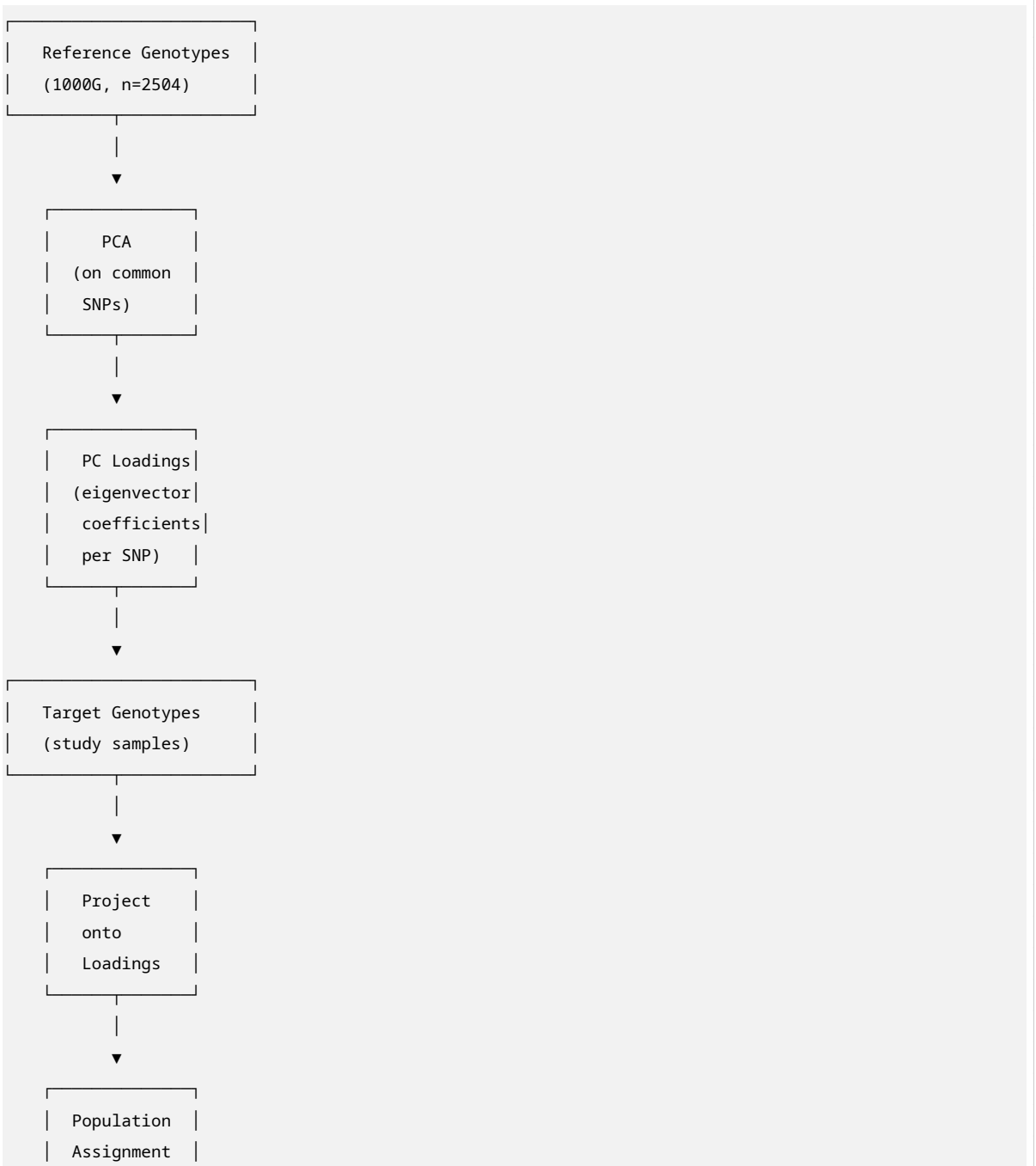
FIG. 15 — LD Correction Workflow

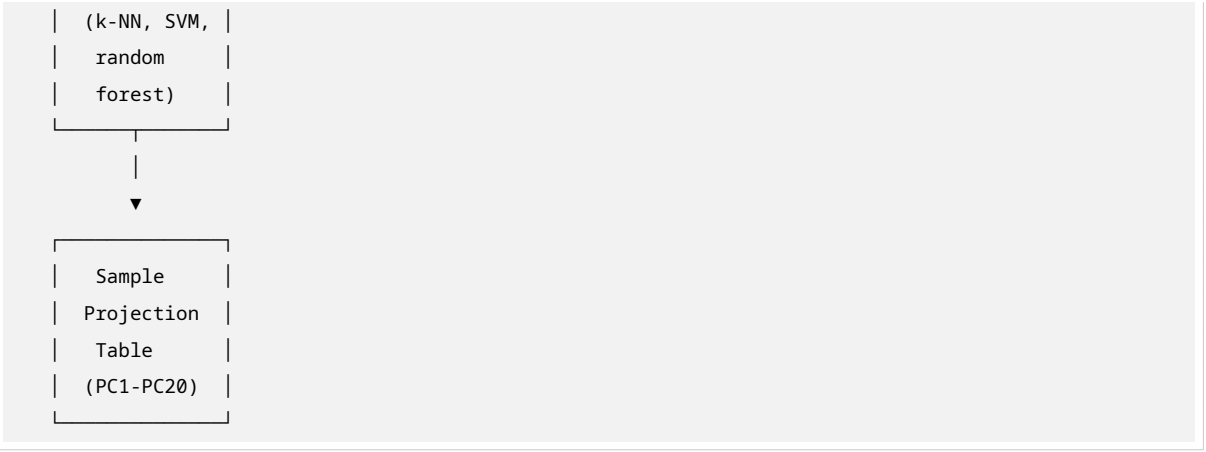




Supporting Tables: Table 15, Table 18

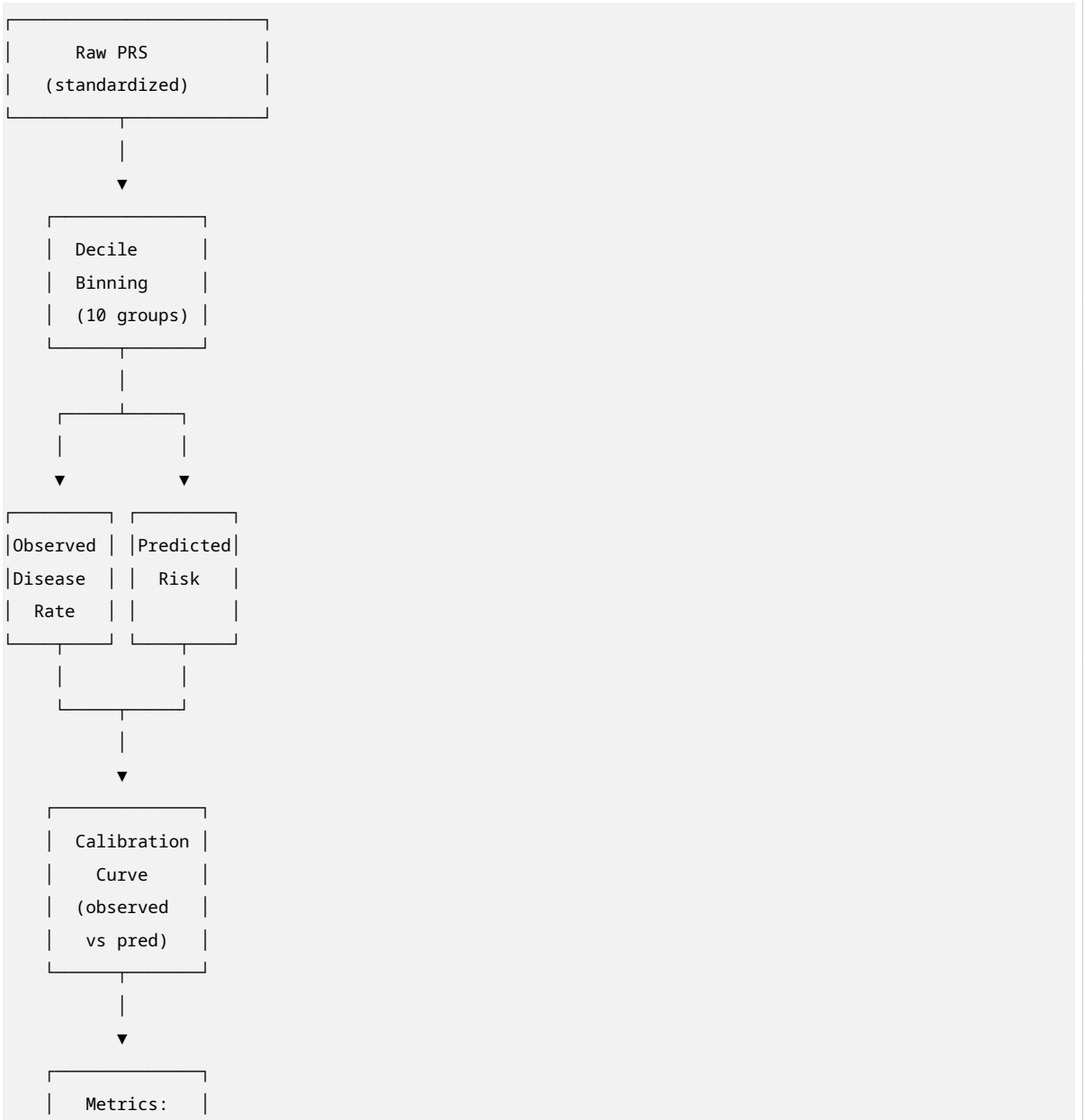
FIG. 16 — Ancestry PCA

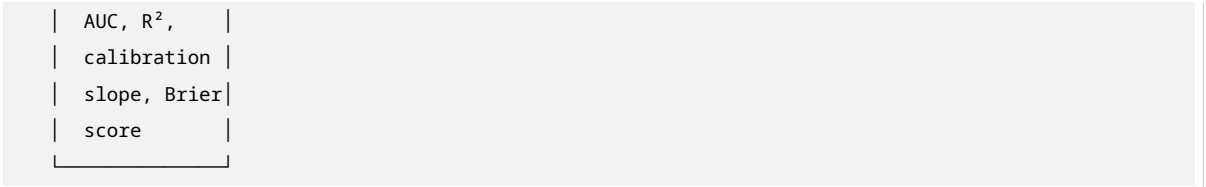




Supporting Tables: Table 19, Table 20

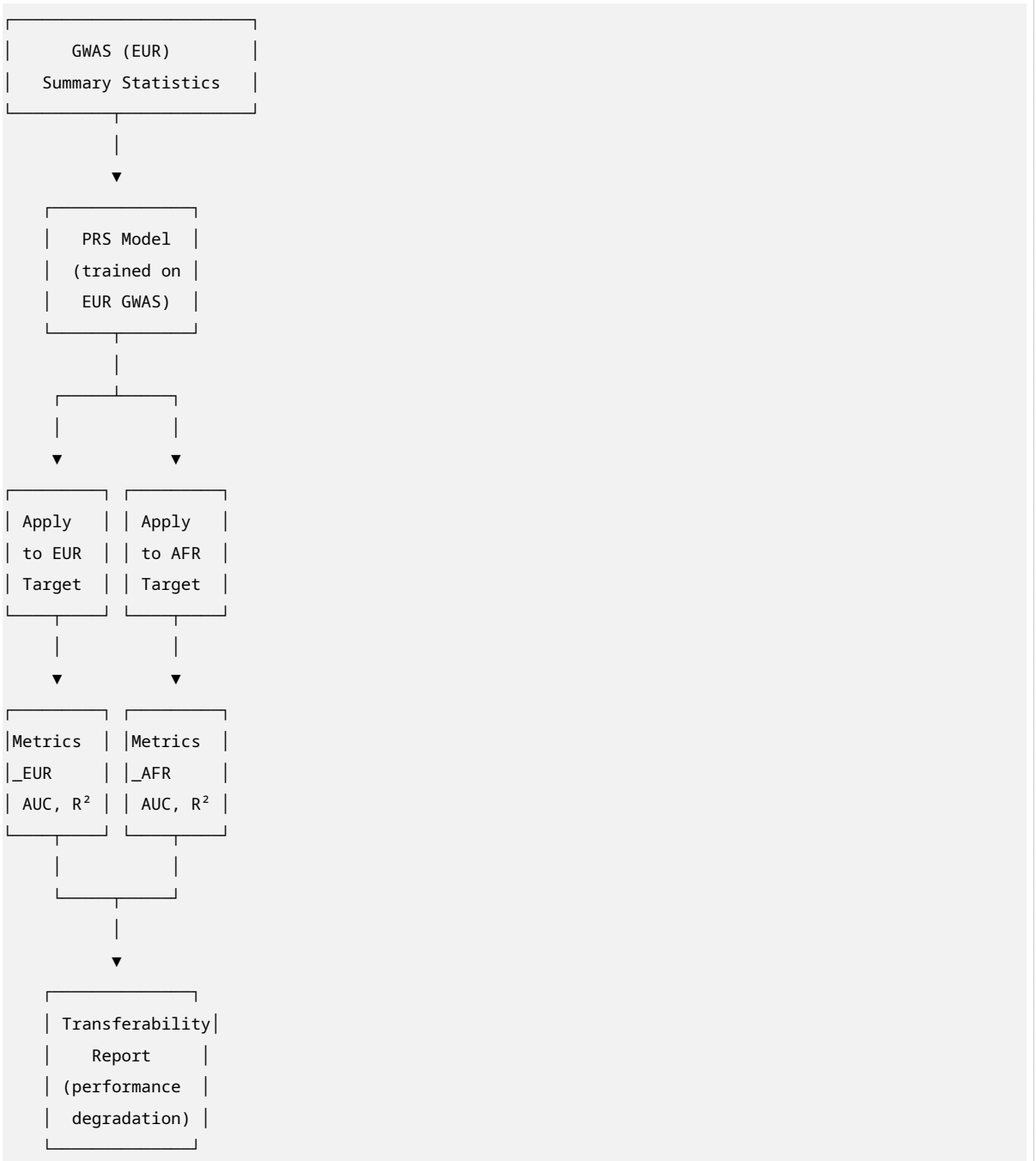
FIG. 17 — PRS Calibration





Supporting Tables: Table 21

FIG. 18 — Population Transferability



Supporting Tables: Table 22

VOLUME II — SUPPORTING TABLES

Table 11: GWAS Study Inventory

Master list of all GWAS studies processed.

Column	Type	Description
study_id	UUID	Unique study identifier
PMID	VARCHAR	PubMed ID
trait	VARCHAR	Phenotype description
population	VARCHAR	Ancestry group
sample_size	INTEGER	Number of individuals
build	VARCHAR	Genome build (e.g., GRCh38)

Table 12: GWAS Summary Statistics Column Map

Standardized column mapping for each GWAS file.

Column	Type	Description
mapping_id	UUID	Primary key
study_id	UUID	FK to Table 11
original_column	VARCHAR	Column name in source file
standard_column	ENUM	chr, pos, ref, alt, beta, se, pval, etc.
transformation	VARCHAR	Any scaling needed (e.g., log OR)

Table 13: GWAS Variant Annotation

Annotation of each variant from GWAS hits.

Column	Type	Description
variant_id	UUID	Unique internal ID
study_id	UUID	FK to study
rsid	VARCHAR	dbSNP identifier
chr	VARCHAR	Chromosome
pos	INTEGER	Position
ref	VARCHAR	Reference allele
alt	VARCHAR	Alternate allele
consequence	VARCHAR	VEP/ANNOVAR consequence

Table 14: Reference Genome Version Registry

Track reference genomes used in the pipeline.

Column	Type	Description
genome_id	UUID	Primary key
species	VARCHAR	e.g., Homo sapiens
build	VARCHAR	GRCh37, GRCh38, T2T
fasta_url	VARCHAR	Source URL
date_ingested	DATE	When fetched

Table 15: LD Reference Panel Catalog

Inventory of LD reference panels used for PRS.

Column	Type	Description
panel_id	UUID	Unique ID
name	VARCHAR	e.g., 1000G Phase 3, UKBB LD
population	VARCHAR	EUR, AFR, EAS, etc.
n_samples	INTEGER	Number of individuals
format	VARCHAR	plink, bgen
source_url	VARCHAR	Source

Table 16: Variant Calling Quality Metrics

Per-sample quality control metrics.

Column	Type	Description
sample_id	VARCHAR	Sample identifier
call_rate	FLOAT	Proportion of variants called
heterozygosity	FLOAT	Observed heterozygosity
ti_tv	FLOAT	Transition/transversion ratio
contamination	FLOAT	Estimated contamination
pass_flag	BOOLEAN	Whether passes QC

Table 17: PRS Calculation Run Registry

Log of every Polygenic Risk Score computation.

Column	Type	Description
prs_run_id	UUID	Unique run
study_id	UUID	GWAS source
panel_id	UUID	LD panel used
method	VARCHAR	PRS-CS, LDpred2, etc.
params	JSONB	Hyperparameters
start_time	TIMESTAMP	Run start

Table 18: PRS Weight Matrix

Final SNP weights per PRS run.

Column	Type	Description
prs_run_id	UUID	FK to run
rsid	VARCHAR	SNP identifier
effect_allele	CHAR(1)	Effect allele
weight	FLOAT	Beta/weight
posterior_prob	FLOAT	Posterior inclusion probability

Table 19: Ancestry PCA Loadings

Principal component loadings from reference populations.

Column	Type	Description
pc_id	UUID	Primary key
panel_id	UUID	Reference panel
pc_number	INTEGER	PC1, PC2, ···
snp	VARCHAR	Variant
loading	FLOAT	Eigenvector coefficient

Table 20: Sample Ancestry Projection

Projected PCs for each target sample.

Column	Type	Description
sample_id	VARCHAR	Sample
pc1	FLOAT	Projected PC1 score
pc2	FLOAT	Projected PC2 score
···	···	up to PC20
assigned_pop	VARCHAR	Inferred superpopulation

Table 21: PRS Calibration Curve Data

Data points for calibration plots.

Column	Type	Description
prs_run_id	UUID	FK
decile	INTEGER	Predicted risk decile
mean_predicted	FLOAT	Mean predicted risk
observed_rate	FLOAT	Observed disease rate
ci_lower	FLOAT	95% CI lower
ci_upper	FLOAT	95% CI upper

Table 22: Population Transferability Statistics

Metrics assessing PRS performance across populations.

Column	Type	Description
source_pop	VARCHAR	GWAS population
target_pop	VARCHAR	Target population
auc	FLOAT	Area under ROC curve
r2	FLOAT	Variance explained
odds_ratio_top_decile	FLOAT	OR for top vs bottom decile

Table 23: Clinical Report Template Mapping

Map PRS results to clinical report sections.

Column	Type	Description
report_section	VARCHAR	e.g., Summary, Risk, Ancestry
data_source	VARCHAR	Table/column providing data
display_type	VARCHAR	Graph, Table, Text
template_ref	VARCHAR	File name of template snippet

[Document continues with Volumes III through VIII, Cross-Cutting Tables, and Appendices…]